

# Why Dataflow Works: The Potato And Candy Explanation

NIConnect 2023

Stephen R. Loftus-Mercer (Aristos Queue)

Principal Software Engineer

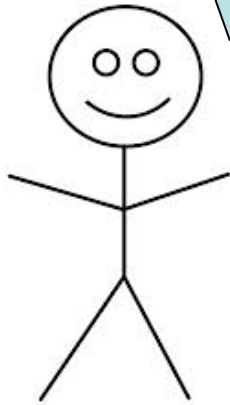
SpaceX

# My background

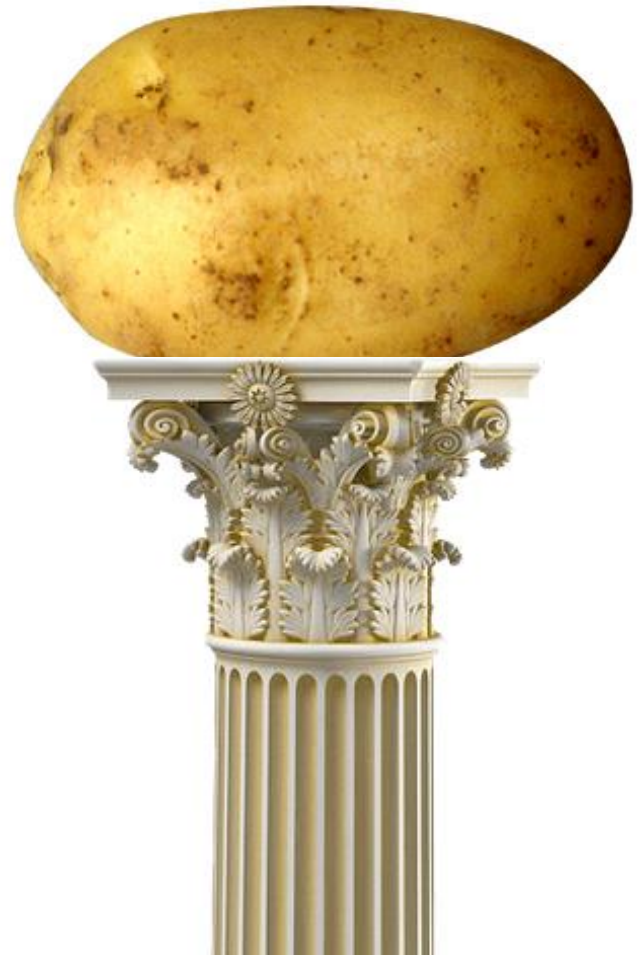
- 22 years in LabVIEW R&D
- SpaceX LabVIEW programmer for Falcon/Dragon Ground Software

# A Potato ~~NOT~~ A Piece of ~~Trash~~ a Pedestal...

What's holding it up in the air? Is it a magic potato?

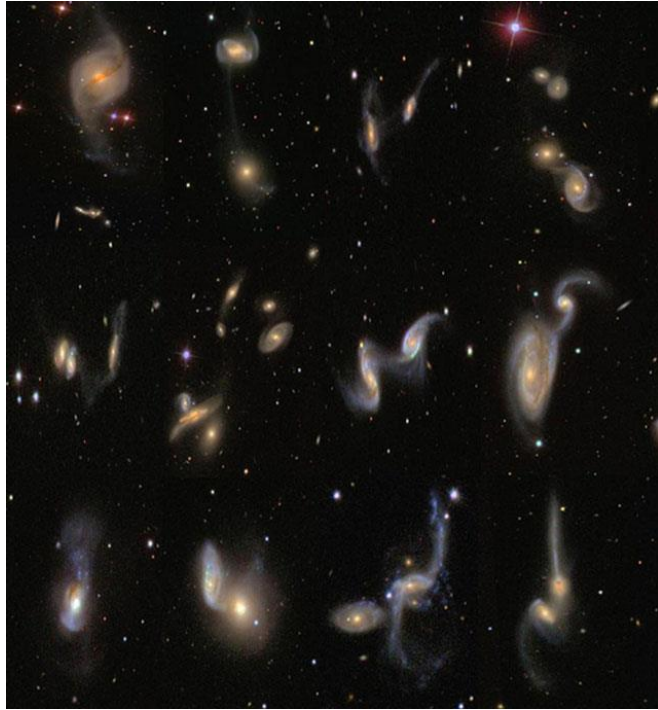


My Nephew, Age 5



# A Really Big Analogy

We can <sup>in</sup> <sub>theory</sub> represent this...

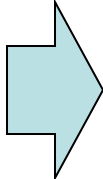


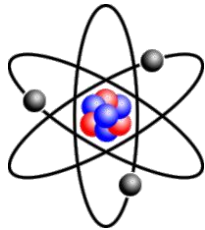
... something like this.

```
0010101011110110101  
0001110101011101001  
1101011110111000010  
1010001000101111010  
1011010001111110100  
01010010110010011...
```

# A Really Big Analogy

So between one atomic clock pulse and the next...

...0111010**1**0111010...  ...0111010**0**0111010...



... imagine a giant computer churning through “ $F=ma$ ” and “ $V=IR$ ” and “ $E=mc^2$ ” for **everything** to compute the next state of the universe and then rendering it on the very large display we call “reality”.

# Theoretical Universe Computer

We can use software to model the universe.  
Can we use the universe to model software?

Two volunteers please.

# Share This Candy Bar



Two more volunteers please.

**Now, Again, Without Breaking It First**





# Romantic? Or Just Cause For Sore Noses?



# Options

- Break In Half
- Simultaneous Access
- Time Sharing
- Morphological Integration

At some level, we are always either breaking down the data or copying it.



Another pair of volunteers please.

# Reading Together



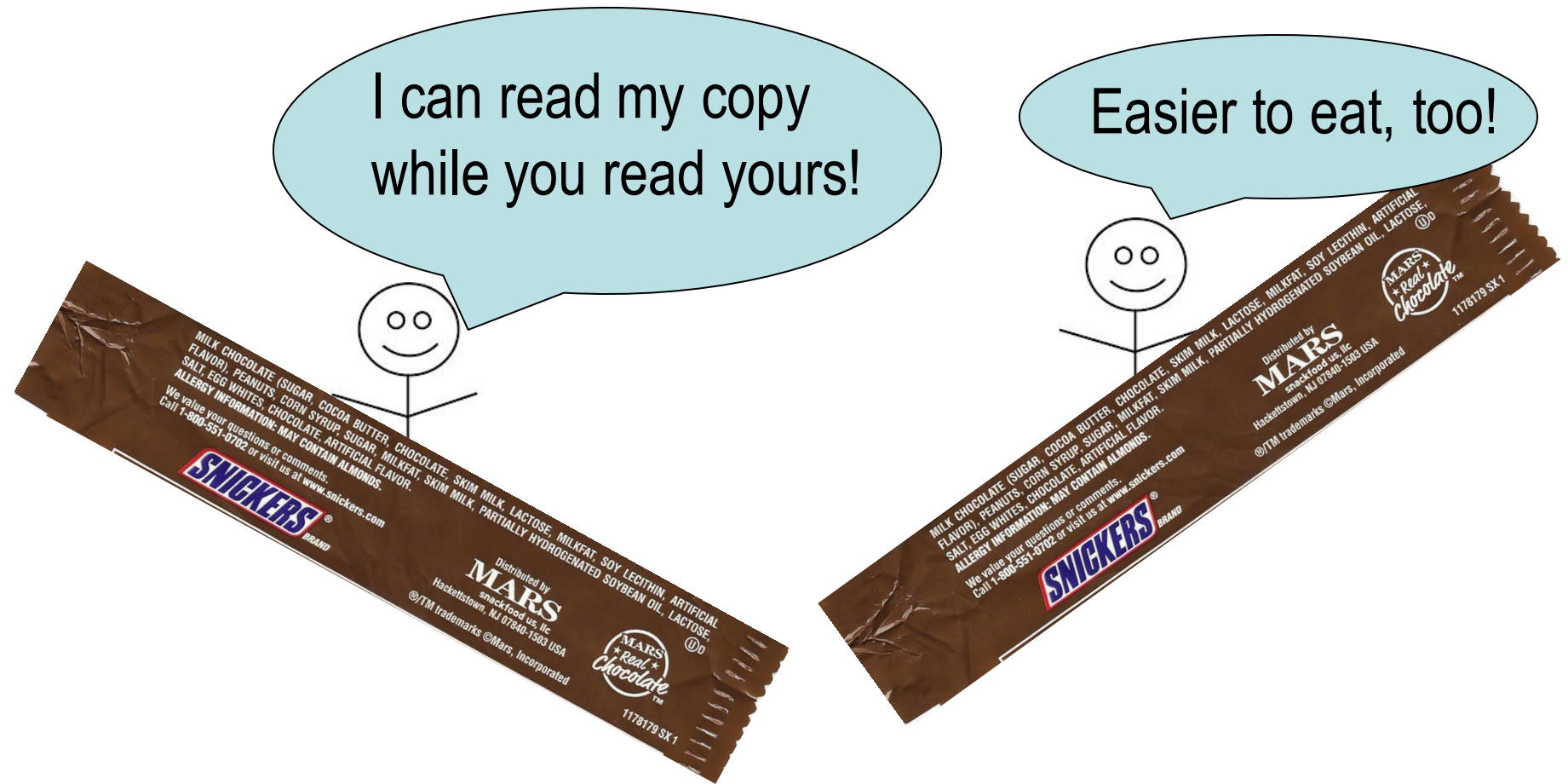
- Heads Close Together
- One Person Reads Aloud
- One Waits For The Other To Finish
- Project On Screen

Who throws away the wrapper? What if the other person is still reading it?

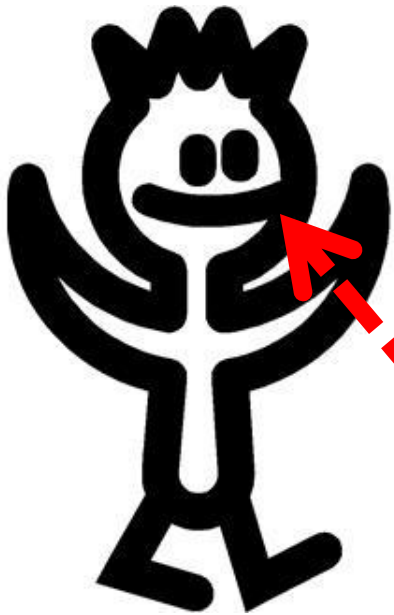
# Easier If Everyone Gets Their Own Copy

I can read my copy while you read yours!

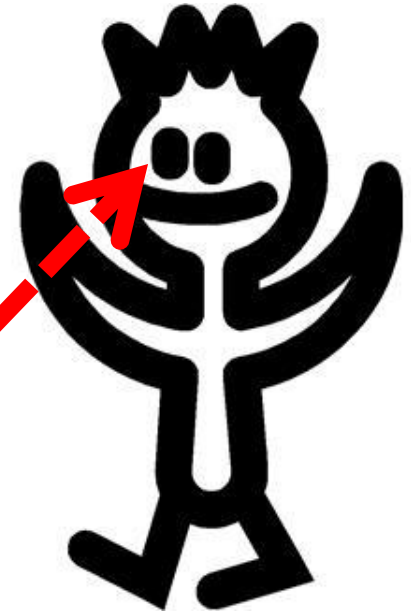
Easier to eat, too!



# All Data Sharing Is Illusion



Not the same photons!



# Dataflow Defined

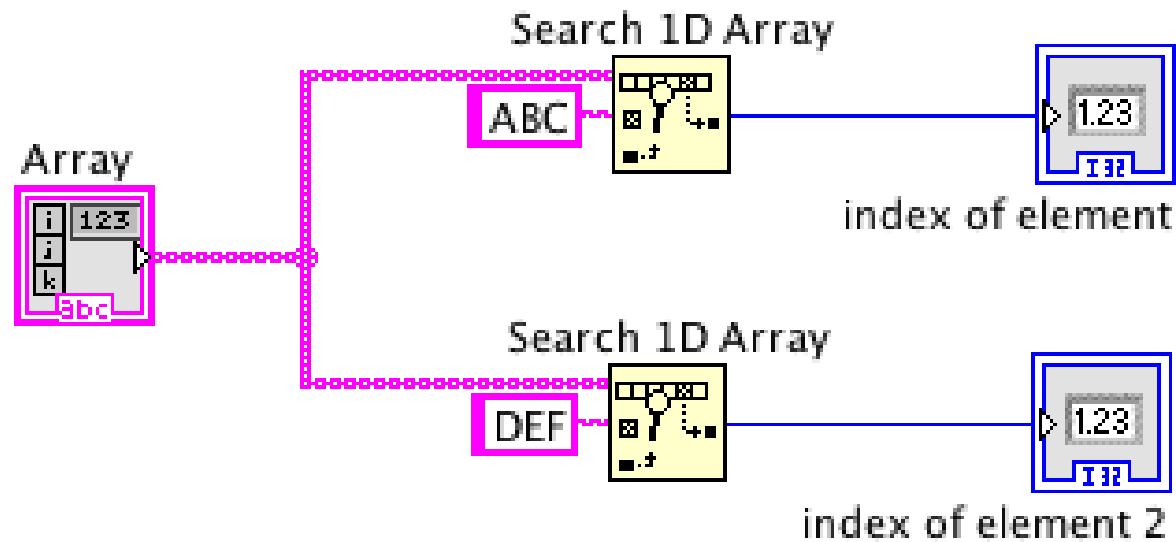
- A model of parallel computing that represents programs as dependence graphs where each operation waits until its input values are available.
- Many models: synchronous, asynchronous, deterministic, and more.

# Dataflow: a guide for the machine

- Who needs the data?
- How soon?
- How much sharing is possible?

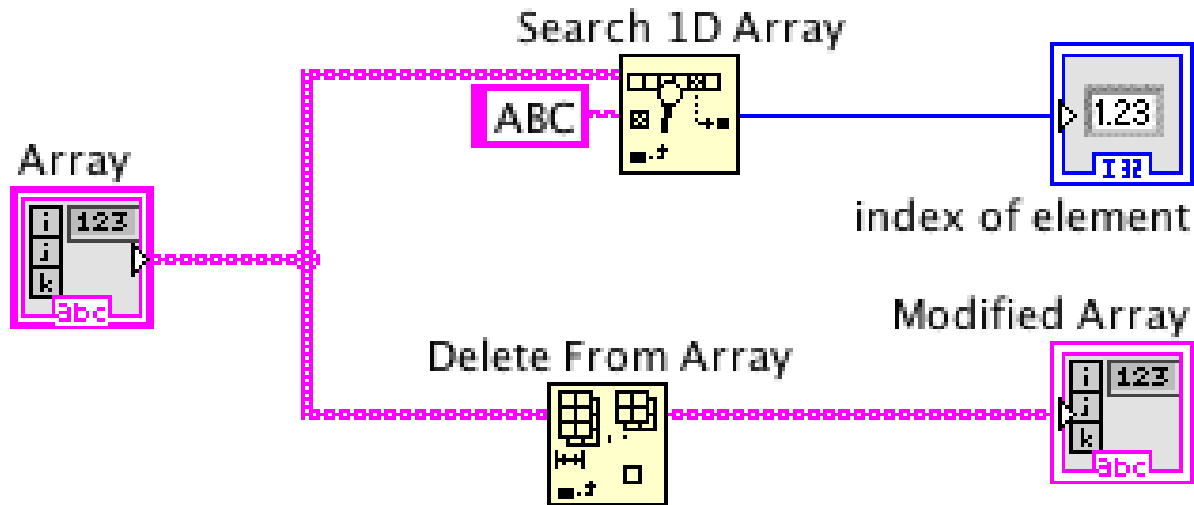
Dataflow declares what the application needs not how those needs should be met. The compiler computes the optimal sharing.

# Two Readers

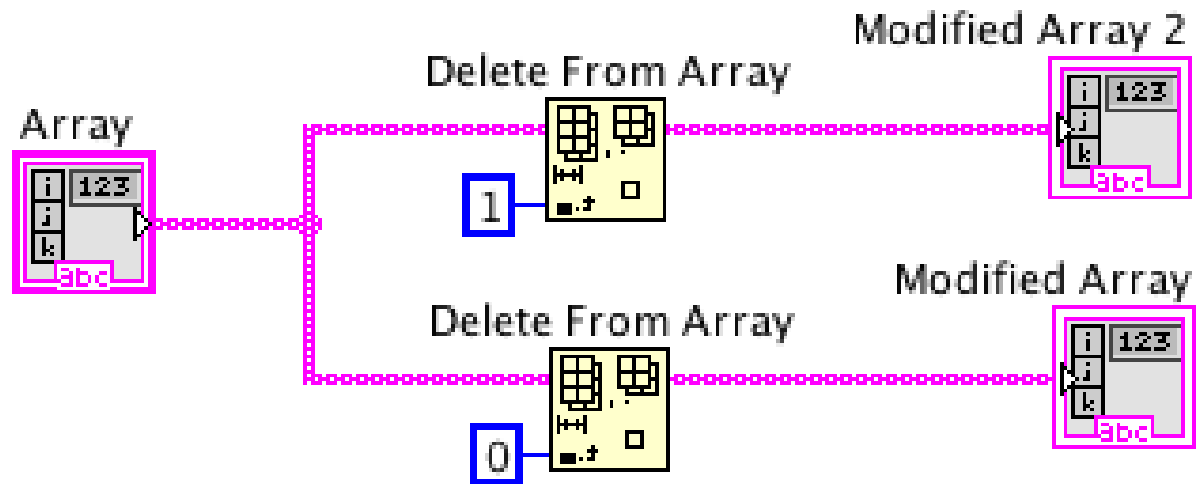




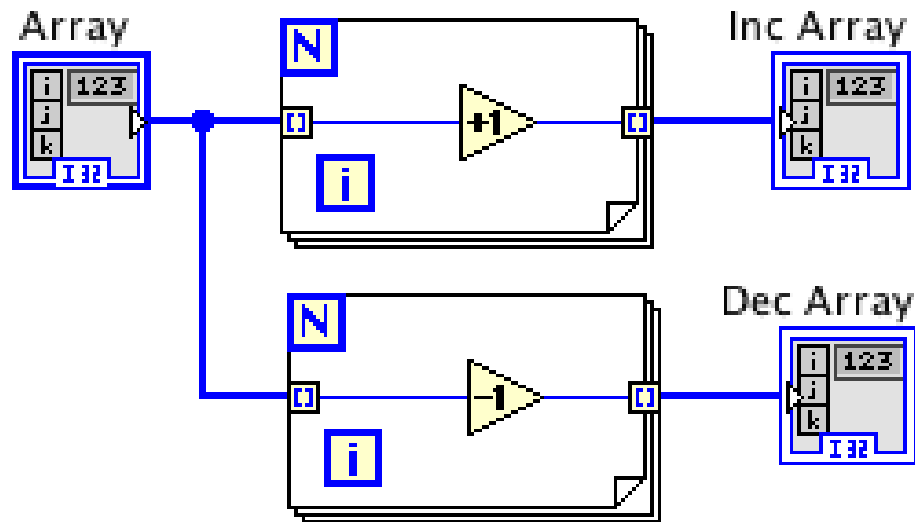
# One Reader, One Writer



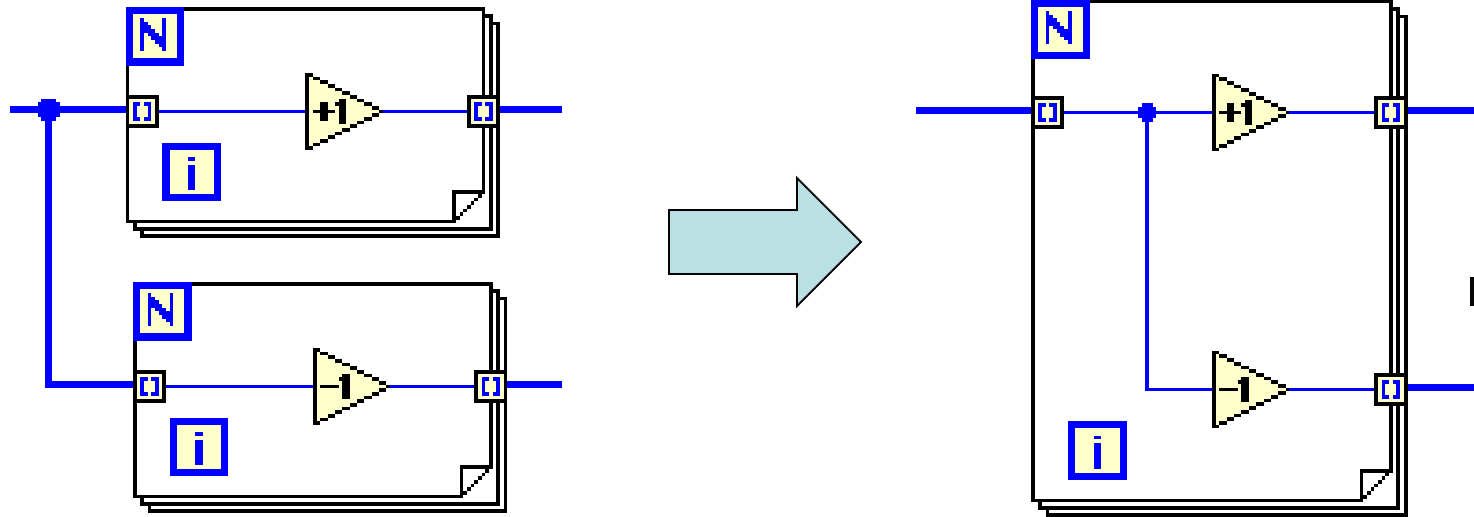
# Two Writers



# Two Writers, Loops



# Two Writers, Loops Fused



(This is morphological integration.)

# Dataflow Works...

... because the compiler prevents you...

- a) from bumping noses as you simultaneously eat both ends of the chocolate and
- b) from pulling the candy out of someone's mouth while they are still chewing and
- c) from buying more candy bars than you need.

# The Never-Ending Quest To Share Data Between Processes

- TCP and UDP
- Network Streams
- Shared Variables x 3
- DMAs
- Web Services
- Peer-to-Peer Streaming
- Queues
- Dynamic Events
- Uninitialized Shift Registers
- RT FIFOs
- Datasocket
- Local Variables
- Programmatic Front Panel Interface
- Target-scoped FIFOs
- Notifier
- Simple TCP/IP Messaging (STM)
- AMC
- HTTP
- FTP
- Global Variables
- Channel Wires
- ... and more

# Definition: “Dataflow Safe”

All data is usable by one and only one process at a time, with clear hand off from one process to the next.

- ✓ All by-value types
- ✓ Channel wires
- ✓ Directed queues
- ✓ Action Engines
- ✓ User Events & Notifiers
- ✓ Unforked references\*
- ✗ Global VIs
- ✗ One-Element Queue Tricks
- ✗ Functional Globals
- ✗ Shared Variables
- ✗ Forked references\*

# Working Within Dataflow

1. Run wires when possible

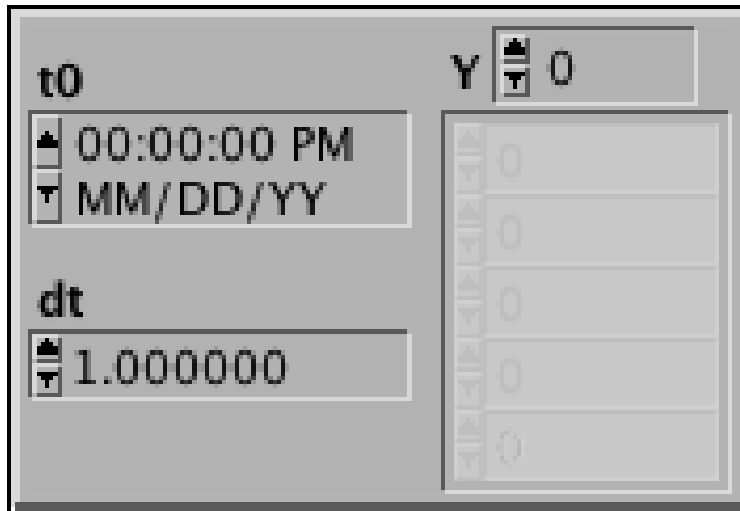




# Working Within Dataflow

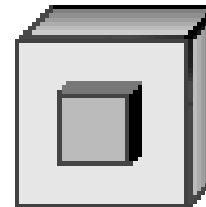
1. Run wires when possible.
2. Execute VIs; pass data.

Waveform



Where is the Run button  
on a control?

LabVIEW Object



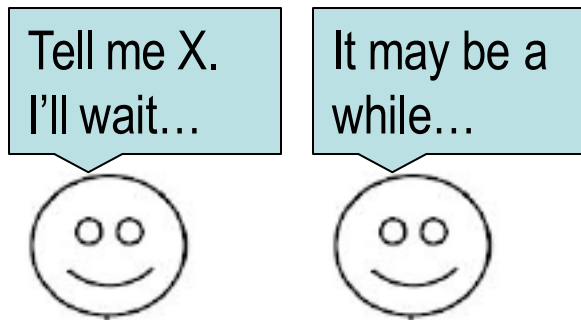
# Working Within Dataflow

1. Run wires when possible.
2. Execute VIs; pass data.

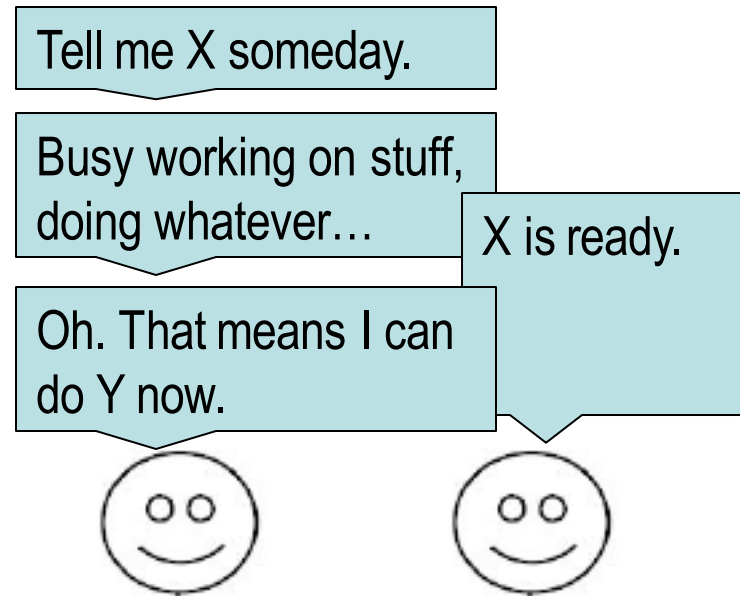


# Working Within Dataflow

1. Run wires when possible.
2. Execute VIs; pass data.
3. Announce instead of request.



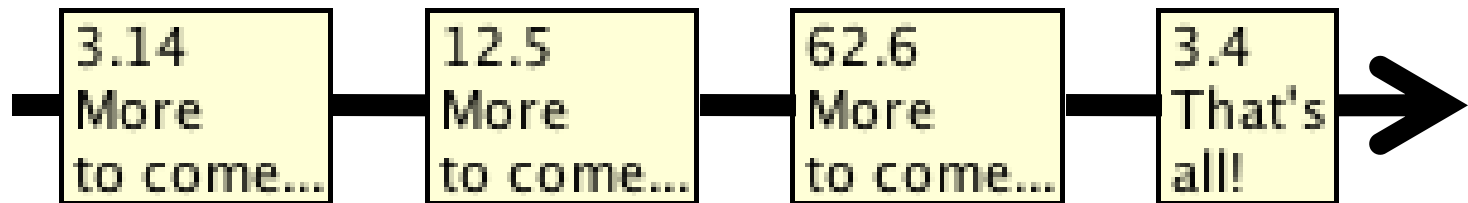
vs.



**Demonstration before point #4...**

# Working Within Dataflow

1. Run wires when possible.
2. Execute VIs; pass data.
3. Announce instead of request.
4. Embed stop signals in communication channel.

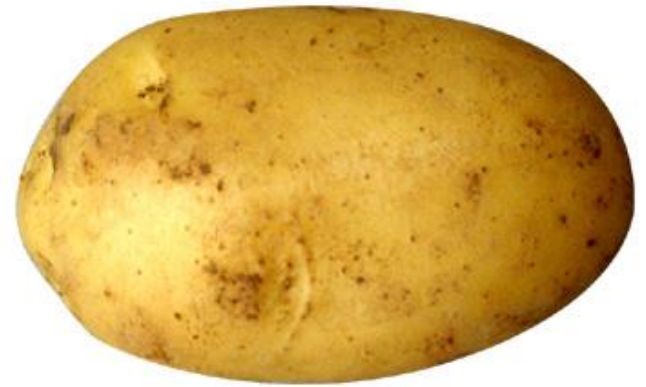


[The Codeless Code, Case 48](#)

# My team is hiring

- <https://spacex.com/jobs>
- “Ground Software Engineer (Falcon & Dragon)”
- “Sr. Ground Software Engineer (Dragon & Falcon)”

# Summary: Wires Are Good!



Any questions?

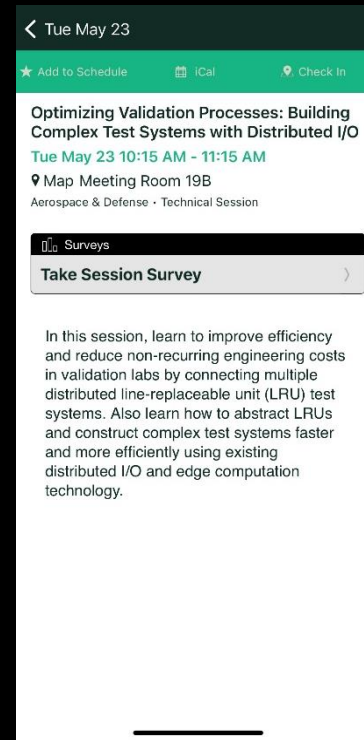
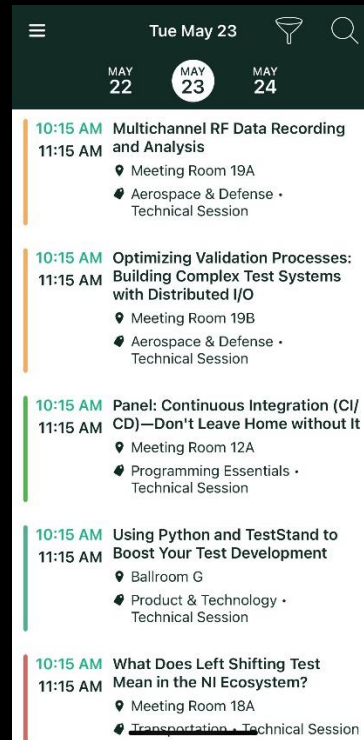
**END OF PRESENTATION**

(stop signal embedded in communications channel)

# Give us your feedback!

## Quick 2 Question Survey

In the mobile app, click into the session you would like to provide feedback for



Click “Take the Session Survey”